Dino Run and Jump

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

6.1 Sviluppi futuri 9

6.2 Considerazioni personali 9

7 Bibliografia 9

7.1 Bibliografia per articoli di riviste: 9

7.2 Bibliografia per libri 9

7.3 Sitografia 10

8 Allegati 10

# Introduzione

## Informazioni sul progetto

* Allievi coinvolti nel progetto: Michea Colautti, Nadia Fasani, Thomas Sartini.
* Classe: I3AA/BB/BC Scuola Arti e Mestieri Trevano.
* Docenti responsabili: Geo Petrini
* Data inizio: 27 gennaio 2022.
* Data di fine: 05 maggio 2022.
* Linguaggio: JavaScript

## Abstract

*We all know the famous Chrome Dino, that little black dinosaur that jumps a lot of cactuses endlessly and tells us that we are not connected to the internet. We've all hated him and loved him at some point in our lives. With this project the Chrome Dino is taken to another level: starting from a previously created project we have improved the user experience and added a new interesting game mechanic. With this project, that’s even multiplayer, the players will be able to jump over cacti not by pressing a button, but by jumping into real life. Using the phone's sensors his dinosaur will jump over obstacles. New features have also been added, such as the possibility to customize the dinosaur and earn rewards. But the project is also available to those who cannot, for one reason or another, jump. For this reason, the phone can also be used, via a button, to control your own dinosaur.*

## Scopo

Lo scopo del progetto è quello di creare una versione multiplayer del famoso Chrome Dino dove gli utenti si possano connettere ad una partita e possano giocare tutti assieme. Il numero di giocatori è quindi variabile, da un minimo di uno ad un massimo di 10. Ci sarà quindi un utente “host” che si occupa di creare la partita e di mostrarla agli altri utenti, idealmente su uno schermo sufficientemente grande. Man mano che si aggiungono giocatori i loro dinosauri appariranno sullo schermo principale.

Il centro del progetto è però il fatto che è possibile far saltare il proprio dinosauro saltando personalmente. Tramite i sensori di movimento del telefono e alla struttura a sessioni del programma i salti verranno trasmessi al server, che comunicherà alla pagina l’evento. Tuttavia per quelle persone che non possiedono un telefono con i giusti sensori oppure sono impossibilitate nel saltare bisogna introdurre una modalità di gioco basilare, come un click, per saltare  
Gli utenti devono poter inoltre poter creare un account tramite il quale potranno personalizzare l’aspetto del prorpio dino e, una volta giocata una o più partite, guadagnarsi delle medaglie che poi saranno visbili in una pagina appoista. Per coloro che non vogliono fare il login saranno creati degli utenti ospiti che verranno eliminati quadno l’utente esce dal sito.

Ci deve essere infine la possibilità di vedere la partita come spettatori, senza interagire con il gioco vero e proprio.

# Analisi

## Analisi del dominio

Non dovremo sviluppare questo progetto da zero, come base avremo infatti il Chrome Dino realizzato da Manuel Grosso (vedi sitografia), nel corso del primo semestre dell’anno scolastico 2021/2022. Questo progetto e quello precedente anno in comune l’aspetto multiplayer, ma per la versione sviluppata da Manuel il numero di giocatori era prescritto a 4. Noi dovremmo rendere questo aspetto dinamico, in modo da permettere maggiore flessibilità. Inoltre per questo progetto la meccanica di salto cambia notevolmente; infatti se prima si premevano dei tasti nel nostro progetto, come già detto, bisognerà saltare veramente. Attualmente non ci sono quindi progetti simili al nostro. Esso non risolve un problema vero e proprio, ma non per questo è poco importante; troviamo infatti molto interessante questa meccanica di gioco, in quanto complica abbastanza un gioco di per sé semplice. Il progetto si rivolge a tutti coloro che hanno voglia di provare qualcosa di nuovo e, avendo introdotto una meccanica di salto alternativa, non esclude coloro che non possono fisicamente saltare o hanno un telefono che non ha i sensori adeguati.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | I giocatori devono poter creare una partita con URL o sessione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Il dinosauro deve essere personalizzabile |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | La GUI deve essere responsive |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Il dinosauro deve saltare sfruttando i sensori del telefono |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto Requisiti** | |
| **001** | Alternativa di gioco in caso di handicap o assenza di sensori. |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Il numero di giocatori deve essere dinamico: da uno a dieci. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Il nome dei giocatori deve apparire a schermo |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Alla fine del gioco deve essere visualizzata una classifica |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto Requisiti** | |
| **001** | Punteggi in ordine decrementale |
| **002** | Visualizzare la/le medaglie direttamente nella classifica |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Al vincitore viene assegnata una medaglia |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto Requisiti** | |
| **001** | Medaglia generata con un algoritmo per rendere il più univoche possibili |

|  |  |
| --- | --- |
| **ID: REQ-09** | |
| **Nome** | È possibile eseguire un login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |  |
| --- | --- | --- |
| **ID: REQ-10** | | |
| **Nome** | | Funzioni aggiuntive di login |
| **Priorità** | | 2 |
| **Versione** | | 1.0 |
| **Note** | |  |
| **Sotto Requisiti** | | |
| **001** | Personalizzazione del dinosauro salvata | |
| **002** | Punteggio salvato nel profilo | |
| **003** | Bacheca per visualizzare le medaglie | |

|  |  |
| --- | --- |
| **ID: REQ-11** | |
| **Nome** | Possibilità di giocare come ospite |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-11** | |
| **Nome** | Possibilità di vedere la partita da remoto |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** |  |

## Use case

Ecco lo *use case* da noi definito:

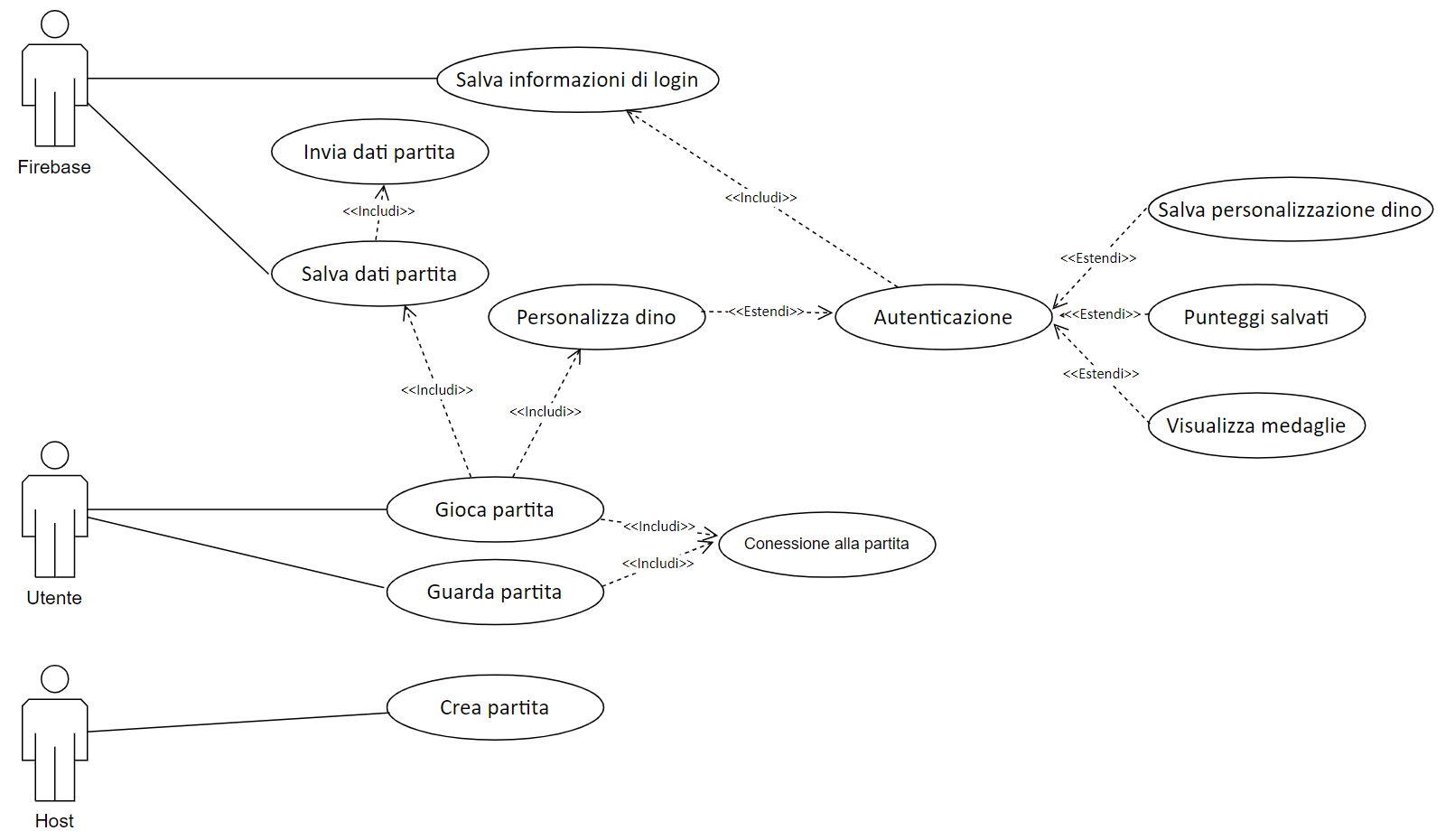


Figura 1 Use cas

## Pianificazione

Per quanto riguarda la pianificazione alleghiamo il diagramma di Gantt iniziale.

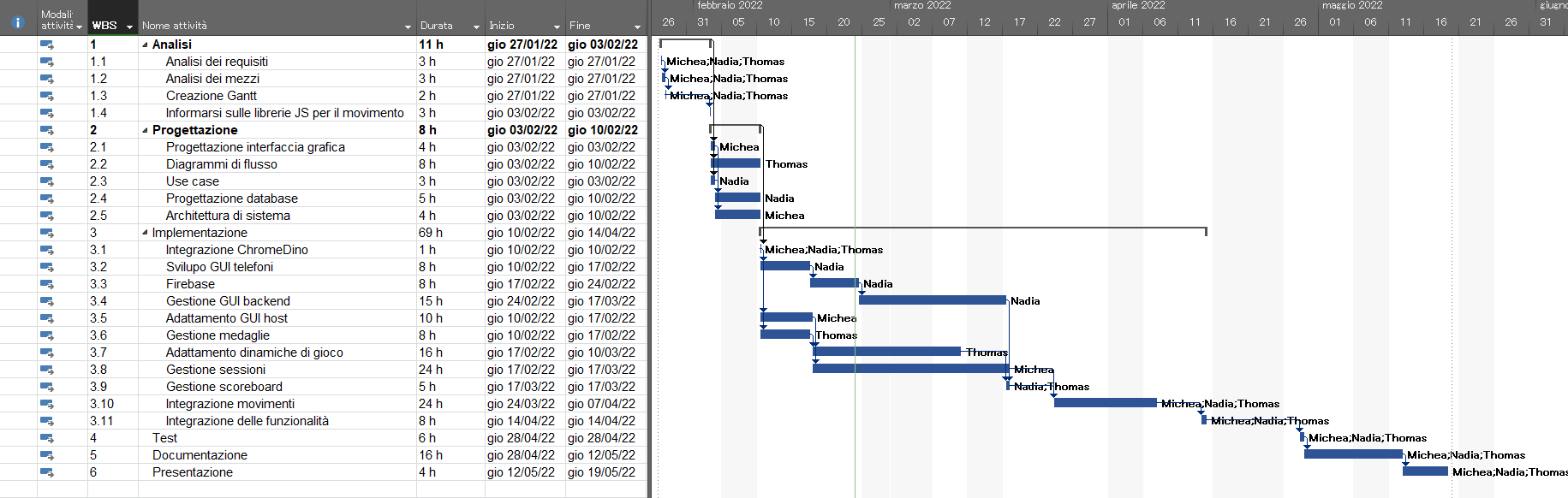


Figura 2 Gantt preventivo

## Analisi dei mezzi

**Software**

* FireBase 8.2.1
* MockFlow 1.4.7
* Draw.io
* VS Code 1.65.2
* GIMP 2.10.24
* GitHub
* GitHub Desktop 2.9.12

**Hardware**

* Laptop personali
* PC scolastici

Come abbiamo già scritto il progetto è scritto in JavaScript, sarà quindi eseguibile da tutti i sistemi operativi. Per quanto riguarda la possibilità di saltare tramite i sensori di movimento invece sui terminali che eseguono IOS non sarà possibile usufruire di questa funzione. Questo a causa delle restrizioni e delle misure di sicurezza applicate da Apple.

# Progettazione

La progettazione è sempre un punto importante in ogni progetto; perciò, abbiamo voluto dedicare tutto il tempo necessario ad essa, definendo tutti gli aspetti che ci sono venuti in mente.

## Design dell’architettura del sistema

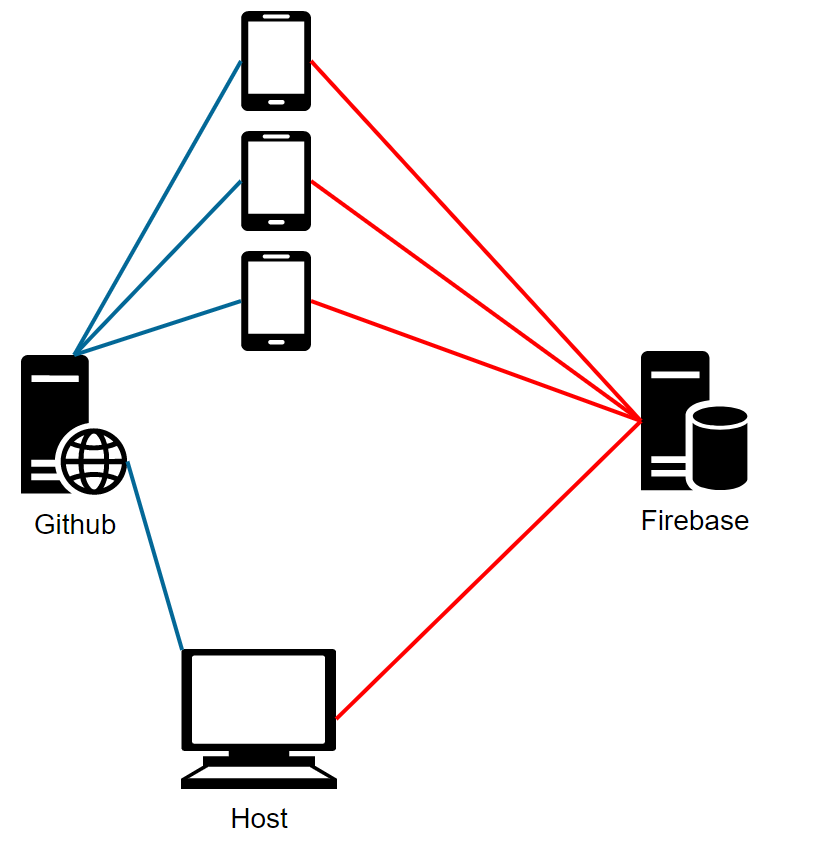


Figura 3 Architettura di sistema

Per l’architettura di sistema abbiamo voluto adottare una struttura abbastanza semplice ma funzionale. Il primo passo della comunicazione, in ordine cronologico, è l’host che crea una partita e la mostra agli utenti. Tutta la parte di gestione del server e delle connessioni è gestita grazie ai server di GitHub. Infatti questa piattaforma offre infatti un servizio chiamato “GitHub Pages”. Esso svolge la funzione di un Web server, togliendo così l’incombenza all’utente.

Contemporaneamente alla creazione della partita l’host comunica al server FireBase le istruzioni necessarie per il buon funzionamento della stessa. Una volta che la partita è stata creata gli utenti si collegano alla pagina, collegandosi quindi ai server GitHub, ma instaurano anche una comunicazione con il server FireBase

## Design dei dati e database

## Design delle interfacce

Per la progettazione delle interfacce abbiamo deciso di dividere i mockups in 2 famiglie: quelle pesate per il telefono e quelle per il computer. Per le interfacce che dovevano essere visualizzate da entrambi i terminali abbiamo realizzato entrambe le versioni.

### Design interfacce telefono:

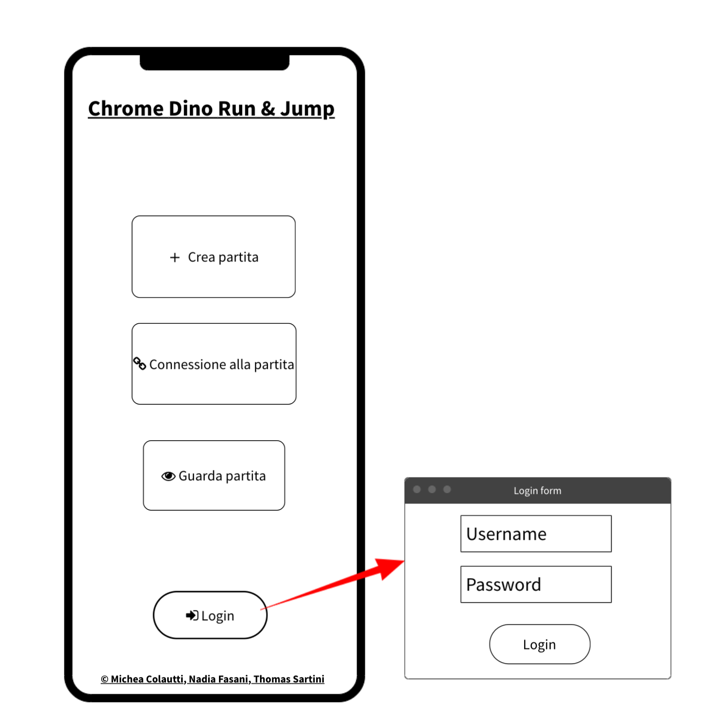


Figura 4 Home page con pop-up di login



Figura Home page con utente loggato

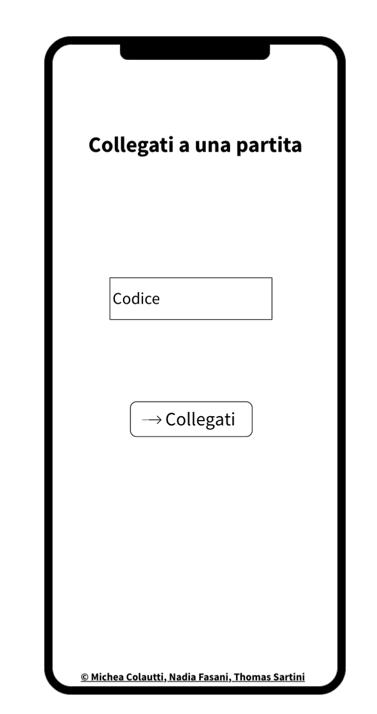


Figura Pagina per collegamento

ad una partita

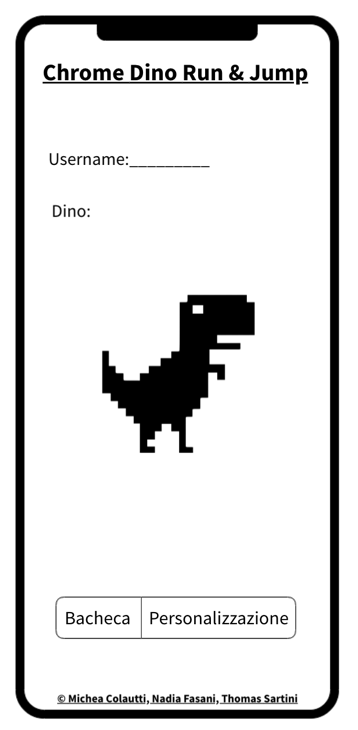


Figura Pagina utente

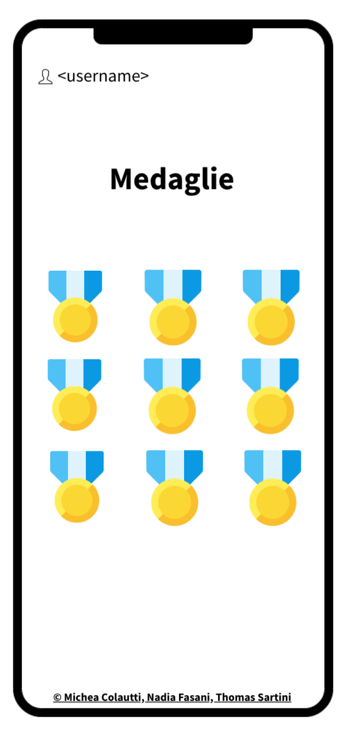


Figura 8 Bacheca medaglie

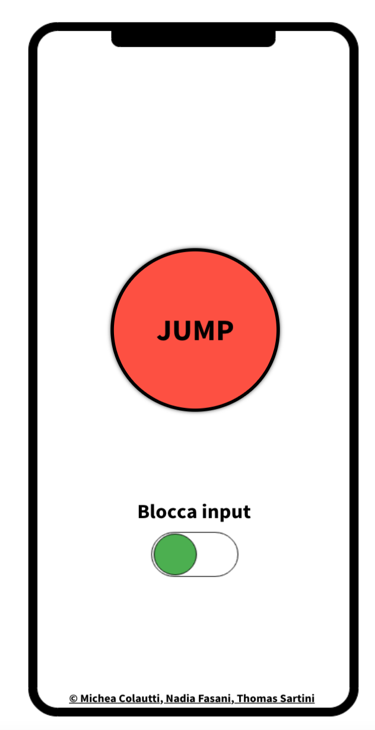


Figura 9 Pagina personalizzazione utente

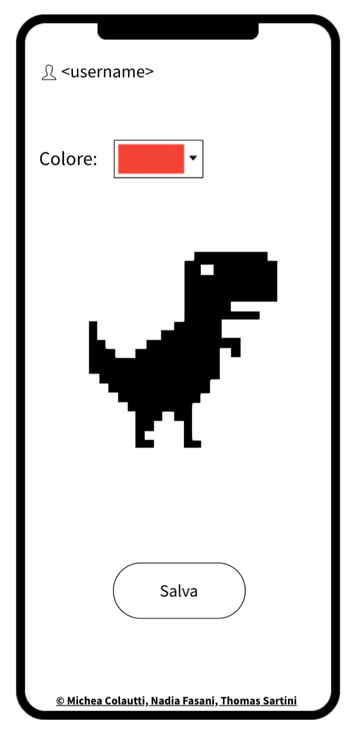


Figura 10 Pagina di gioco

### Design interfacce computer

Figura Home page

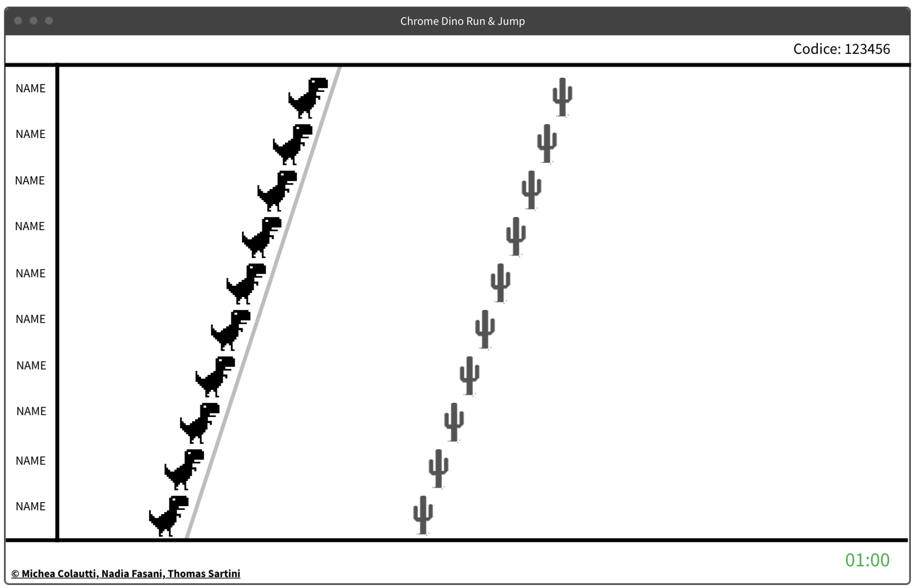


Figura Creazione di una partita



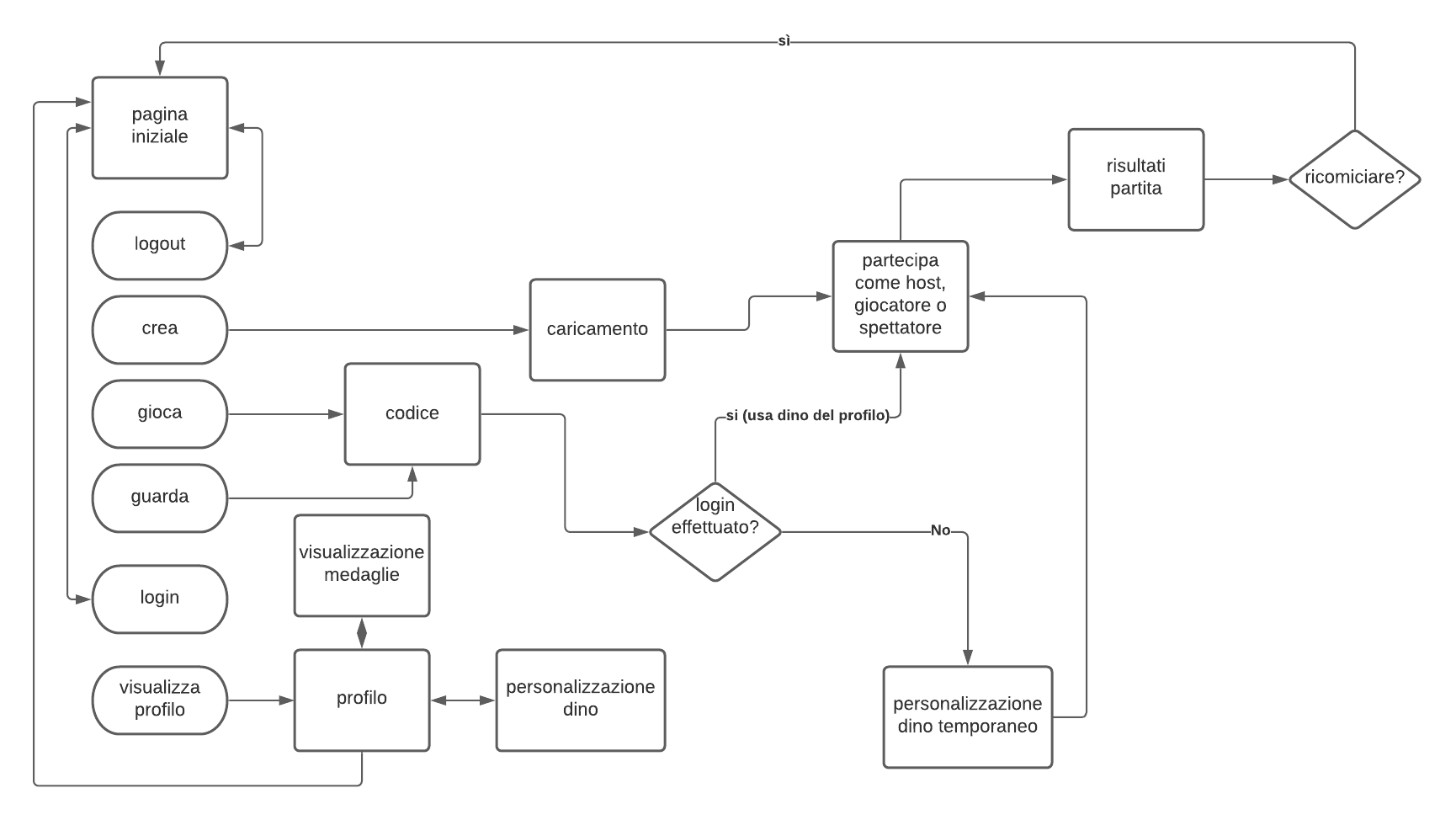
Figura Collegamento ad una partita

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura Classifica

## Design procedurale

Per quanto riguarda il design procedurale alleghiamo il diagramma di flusso da noi pensato.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …